ReLU & Atlas LSH Neural Networks:

Dynamic Parameter Selection Beyond ReLU

1. Introduction

A standard feedforward neural network is often described as a fixed sequence of matrix multiplications and nonlinearities:

$$h_{l+1} = \sigma(W_l h_l + b_l)$$

However, this description hides an important fact: in networks using ReLU nonlinearities, the effective parameter selection and information route is already **input dependent**.

For a given input, some neurons are active and others inactive. The network therefore selects only a subset of its parameters for use.

This observation motivates a broader view:

> Neural networks can be regarded as systems that dynamically select computational operators and parameters in response to input.

Atlas LSH neural networks extend this idea by replacing local activation decisions with explicit routing decisions generated by locality sensitive hashing (LSH).

---

2. ReLU Networks as Decision Matrix Systems

Consider a ReLU layer:

$$h_{l+1} = \mathrm{ReLU}(W_l h_l)$$

Define a diagonal decision matrix

$$D_l(x) = \begin{bmatrix} d_1 & 0 & \cdots \\ 0 & d_2 & \cdots \\ \vdots & & \ddots \end{bmatrix}$$

where

$$d_i = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$h_{l+1} = D_l(x) W_l h_l$$

The matrix D_l depends on the input.

Rows of W_l corresponding to inactive neurons are effectively removed

and columns of the following layer are similarly disabled.

Thus ReLU networks already perform:

- input-dependent weight selection,
- dynamic subnetwork extraction,
- conditional computation.

The active network for an input is

$$W_L D_{L-1} W_{L-1} \cdots D_1 W_1$$

which defines one linear region of a piecewise linear function.

---

3. Replacing Local Decisions with Global Decisions

ReLU decisions are:

- local,
- neuron-specific,
- learned indirectly.

Atlas LSH networks instead compute a global routing code:

$$b(x) = \mathrm{LSH}(x)$$

where

$$b(x) \in \{0,1\}^k$$

is a binary hash.

Each bit or group of bits controls parameter selection.

Conceptually:

$$h_{l+1} = F(h_l, b(x))$$

rather than

$$h_{l+1} = F(h_l, D_l(h_l))$$

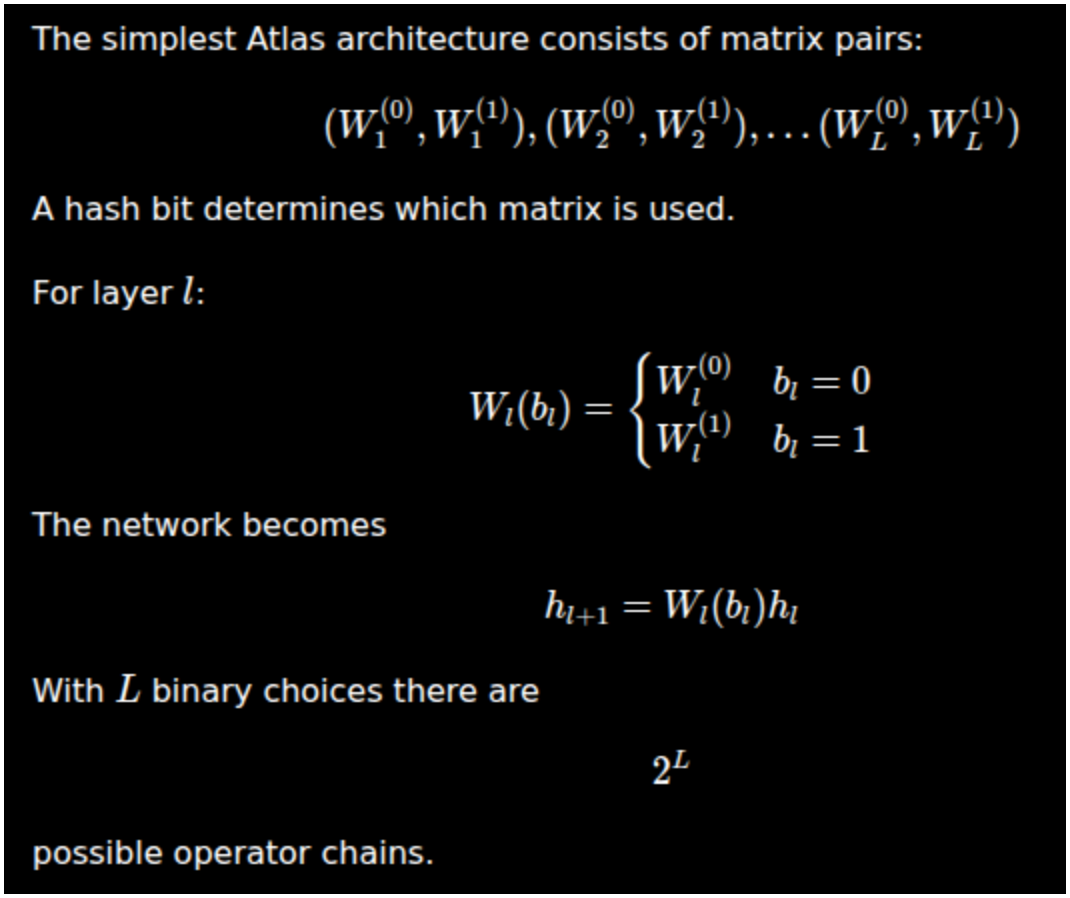
$b(x) \in \{0,1\}^k$ is a binary hash.

Each bit or group of bits controls parameter selection.

The LSH code acts as an external decision system controlling which parameters participate in computation.

---

4. Matrix Pair Toy Model


The simplest Atlas architecture consists of matrix pairs:
$$(W_1^{(0)}, W_1^{(1)}), (W_2^{(0)}, W_2^{(1)}), \dots (W_L^{(0)}, W_L^{(1)})$$
A hash bit determines which matrix is used.
For layer l:
$$W_l(b_l) = \begin{cases} W_l^{(0)} & b_l = 0 \\ W_l^{(1)} & b_l = 1 \end{cases}$$
The network becomes
$$h_{l+1} = W_l(b_l) h_l$$
With L binary choices there are
$$2^L$$
possible operator chains.

A network with only 32 matrix pairs already defines

$2^{32} \approx 4.3*10^9$

distinct computational paths.

This toy model illustrates how enormous combinatorial diversity can arise from a small number of routing decisions.

---

5. Decision Granularity

A key design choice is:

> What does a routing decision control?

Many possibilities exist.

5.1 Whole-Matrix Selection

One bit selects one matrix from a pair.

$$W_l = W_l^{(b)}$$

Advantages:

- conceptually simple,
- easy to implement,
- large operator diversity.

Disadvantages:

- coarse control,
- parameter cost grows quickly.

5.2 Matrix Bank Selection

A group of bits selects one matrix from a larger bank.

$$W_l = W_l^{(i)}$$

where

$$i = \text{decode}(b_{a:b})$$

For m bits:

$$2^m$$

candidate matrices are available.

Example:

8 bits → 256 candidate operators.

5.3 Submatrix Selection

Matrices are partitioned into blocks:

$$W = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

LSH bits determine which blocks are active.

Examples:

- row blocks,
- column blocks,
- tiles,
- attention heads,
- channel groups.

This resembles structured sparsity.

5.4 Row/Column Gating

Hash bits select rows:

$$W' = D_r(b)W$$

or columns:

$$W' = WD_c(b)$$

where D_r, D_c are hash-controlled binary masks.

This is conceptually close to ReLU gating except decisions originate from global hash codes.

5.5 Weight Group Selection

Parameters are divided into groups:

Parameters are divided into groups:

$$W = \sum_i g_i(b)W_i$$

where

$$g_i(b) \in \{0, 1\}$$

determines whether group i contributes.

This yields sparse conditional computation.

5.6 Additive Parameter Insertion

Rather than replacing a matrix, LSH selects additive components:

$$W_{\text{eff}} = W_{\text{base}} + \Delta W(b)$$

where

$$\Delta W(b)$$

comes from a hash-indexed parameter table.

This resembles:

- adapters,
- LoRA,
- hypernetworks,
- product-key memories.

5.7 Operator Composition

Bits can select operators rather than weights.

Examples:

$$W \to \begin{cases} \text{dense} \\ \text{low rank} \\ \text{FFT} \\ \text{attention} \\ \text{identity} \end{cases}$$

The (perhaps learned or evolved) Atlas becomes a computational graph.

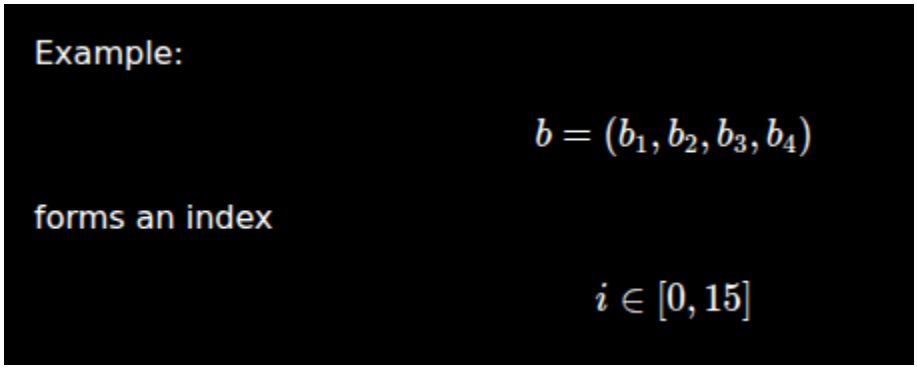
---

6. Using Bit Clumps

Single bits provide binary choices.

Groups of bits provide exponentially many choices.

Example:



used to retrieve:

- matrices,
- tensor blocks,
- adapters,
- routing coefficients,
- expert identities.

A k-bit hash provides

2^k

possible addresses.

This connects Atlas networks naturally to:

- associative memories,
- product-key memories,
- mixture-of-experts systems,
- learned caches.

---

7. Local versus Global Routing

Property	**ReLU**	**Atlas LSH**
Decision source	Local activations	Global hash

Decision type	Threshold	Hash lookup
Routing granularity	Neuron	Arbitrary
Regions	Piecewise linear	Quantized hash cells
Parameter selection	Implicit	Explicit
Controllability	Limited	High

ReLU boundaries move continuously during training.

LSH boundaries are quantized by hash bit transitions.

Consequently Atlas networks may exhibit more jagged decision regions while simultaneously offering far richer explicit parameter selection.

---

8. The Atlas Perspective

The central idea is simple:

> A neural network need not use one fixed parameter set. Instead, an input may first locate itself within an atlas of regions, and the atlas determines which computational machinery is deployed.

The atlas may choose:

- entire matrices,
- matrix blocks,
- parameter groups,
- operators,
- adapters,
- experts,
- memory entries.

Even modest routing structures can produce astronomical numbers of distinct computational pathways.

Viewed this way, Atlas LSH neural networks generalize the dynamic parameter selection already present in ReLU networks, replacing local activation gates with explicit, globally coordinated routing decisions.